

Práctica 4 – Contenedores

Descripción de los pasos seguidos para cumplir los objetivos

Actividad 1:

Crear un contenedor con docker que contenga nuestra aplicación que permita comprobar su funcionamiento (e.g. una pagina web)

Para esto, debemos crear nuestra página web(HTML) y nuestro archivo Dockerfile:

```
index.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Mi Página Web de Docker</title>
5  </head>
6  <body>
7  |   <h1>ESTA ES MI PAGINA WEB DE DOCKER</h1>
8  </body>
9  </html>
10
```

```
Dockerfile > ...
1  # Usa una imagen base que contenga un servidor web --> nginx
2  FROM nginx:latest
3
4  # Copia el archivo index.html al directorio donde nginx sirve los archivos estáticos
5  COPY index.html /usr/share/nginx/html/index.html
```

Actividad 2:

Crear un repositorio en ECR y subir el contenedor creado en el paso 1

1. Primero debemos crear nuestro repositorio ECR Privado en AWS. Solo necesitamos ponerle un nombre.
2. A continuación debemos poner nuestras credenciales en la consola AWS CLI de nuestro ordenador, rellenando el archivo '~/.aws/credentials' con los siguiente encontrado en CLI > AWS Details > Cloud Access:

```
[default] aws_access_key_id=<Access_Key>
aws_secret_access_key=<Secret_Key>
```

aws_session_token=<Token>

3. A continuación debemos seguir los siguientes pasos para verificar nuestro Docker, construir la imagen, añadir el TAG a la imagen y finalmente hacer el PUSH a nuestro repositorio ECR.

Si seleccionamos nuestro Repositorio y le damos a 'View push commands' nos mostrará los pasos aplicados a nuestro repositorio:

1. Retrieve an authentication token and authenticate your Docker client to your registry.

Use the AWS CLI:

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 329187668450.dkr.ecr.us-east-1.amazonaws.com
```

Note: if you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.

2. Build your Docker image using the following command. For information on building a Docker file from scratch, see the instructions [here](#). You can skip this step if your image has already been built:

```
docker build -t webserver_cn .
```

3. After the build is completed, tag your image so you can push the image to this repository:

```
docker tag webserver_cn:latest 329187668450.dkr.ecr.us-east-1.amazonaws.com/webserver_cn:latest
```

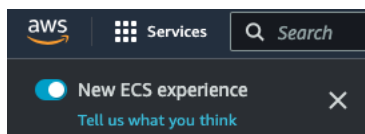
4. Run the following command to push this image to your newly created AWS repository:

```
docker push 329187668450.dkr.ecr.us-east-1.amazonaws.com/webserver_cn:latest
```

Actividad 3:

Desplegar el contenedor usando ECS

Para esto usaremos la antigua interfaz de AWS haciendo click en:



****SOLO DISPONIBLE ANTES DEL 4 DE DICIEMBRE****

The classic Amazon ECS console is reaching the end of life and will no longer be available after December 4, 2023. We recommend that you switch immediately to the new Amazon ECS console for a better experience. To learn more about the new Amazon ECS console experience, visit the [documentation page](#), and you can also review and follow the new Amazon ECS console roadmap on [GitHub](#).

1. Primero debemos de crear el Cluster de EC2

Select cluster template

The following cluster templates are available to simplify cluster creation. Additional configuration and integrations can be added later.

Networking only ⓘ

Resources to be created:

Cluster
VPC (optional)
Subnets (optional)

ⓘ For use with either AWS Fargate (Windows/Linux)
or with External instance capacity.

EC2 Linux + Networking

Resources to be created:

Cluster
VPC
Subnets
Auto Scaling group with Linux AMI

EC2 Windows + Networking

Resources to be created:

Cluster
VPC
Subnets
Auto Scaling group with Windows AMI

*Required

Cancel

Next step

Configure cluster

Cluster name* ⓘ

☐ Create an empty cluster

Instance configuration

Provisioning Model ☒ On-Demand Instance

With On-Demand Instances, you pay for compute capacity by the hour, with no long-term commitments or upfront payments.

☐ Spot

Amazon EC2 Spot Instances let you take advantage of unused EC2 capacity in the AWS cloud. Spot Instances are available at up to a 90% discount compared to On-Demand prices. [Learn more](#)

EC2 instance type* ⓘ ⓘ
☐ Manually enter desired instance type

Number of instances* ⓘ

EC2 AMI ID* ⓘ



Root EBS Volume Size (GiB) ⓘ

Key pair ⓘ ⓘ


You will not be able to SSH into your EC2 instances without a key pair. You can create a new key pair in the [EC2 console](#).

Networking

Configure the VPC for your container instances to use. A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You can choose an existing VPC, or create a new one with this wizard.

VPC vpc-058cda2ce5f98b8a...  


Check the structure for [vpc-058cda2ce5f98b8ab](#) in the Amazon EC2 console.


Subnets 

subnet-098581d79edb517d1
(172.31.0.0/20) - us-east-1
assign ipv6 on creation: Disabled

subnet-046b25d65808e65e7
(172.31.32.0/20) - us-east-1
assign ipv6 on creation: Disabled

Select a subnet...

Auto assign public IP Enabled 

Security group sg-01c0d70cc7804756...  


Rules for [sg-01c0d70cc78047564](#) in the EC2 Console.

Container instance IAM role

The Amazon ECS container agent makes calls to the Amazon ECS API actions on your behalf, so container instances that run the agent require the `ecsInstanceRole` IAM policy and role for the service to know that the agent belongs to you. If you do not have the `ecsInstanceRole` already, we can create one for you.

Container instance IAM role LabRole 

Tras crearlo, veremos como se nos crea una Instancia y Security Group. Veremos como se nos asocia la instancia a nuestro cluster gracias al Security Group.

ECS Instance - ... I-Oa38eb188ad6fc1c6  Running

2. A continuación crearemos el 'TASK'

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

FARGATE



Price based on task size

Requires network mode awsvpc

AWS-managed infrastructure, no Amazon EC2 instances to manage

EC2



Price based on resource usage

Multiple network modes available

Self-managed infrastructure using Amazon EC2 instances

EXTERNAL



Price based on instance-hours and additional charges for other AWS services used

Self-managed on-premise infrastructure with ECS
Anywhere

Task definition name* ⓘ

Requires compatibilities* EC2

Task role ⓘ

Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the [IAM Console](#) ⓘ

Network mode ⓘ

If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. Windows tasks support the <default> and awsvpc network modes.

Task execution IAM role

This role is required by tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the ecsTaskExecutionRole already, we can create one for you.

Task execution role ⓘ

Task size ⓘ

The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 or External launch type. Container level memory settings are optional when task size is set. Task size is not supported for Windows containers.

Task memory (MiB)

No Asignar Nada

The amount of memory (in MiB) used by the task. It can be expressed as an integer using MiB, for example 1024, or as a string using GB, for example '1GB' or '1 gb'.

Task CPU (unit)

The number of CPU units used by the task. It can be expressed as an integer using CPU units, for example 1024, or as a string using vCPUs, for example '1 vCPU' or '1 vcpu'.

Task memory maximum allocation for container memory reservation



Task CPU maximum allocation for containers



Y asociamos nuestro ECR a la 'TASK'

▼ Standard

Container name*	<input type="text" value="webserver_cn"/>							
Image*	<input type="text" value="329187668450.dkr.ecr.us-east-1.amazonaws.com/webserver_cn:latest"/>							
Private repository authentication*	<input type="checkbox"/>							
Memory Limits (MiB)*	<div><div>Hard limit ▼</div><div>120 300</div></div> <div>+ Add Soft limit<p>Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the `memory` and `memoryReservation` parameters, respectively, in task definitions. ECS recommends 300-500 MiB as a starting point for web applications.</p></div>							
Port mappings	<table><thead><tr><th>Host port</th><th>Container port</th><th>Protocol</th></tr></thead><tbody><tr><td><input type="text" value="80"/></td><td><input type="text" value="80"/></td><td><div>tcp ▼</div></td></tr></tbody></table> <div>+ Add port mapping</div>	Host port	Container port	Protocol	<input type="text" value="80"/>	<input type="text" value="80"/>	<div>tcp ▼</div>	<div></div> <div></div>
Host port	Container port	Protocol						
<input type="text" value="80"/>	<input type="text" value="80"/>	<div>tcp ▼</div>						

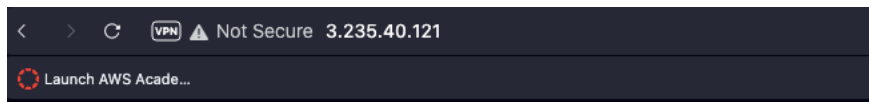
3. Finalmente correremos nuestra 'TASK' previamente creada

Run Task

Select the cluster to run your task definition on and the number of copies of that task to run. To apply container overrides or target particular container instances, click Advanced Options.

Launch type	<input type="radio"/> FARGATE <input checked="" type="radio"/> EC2 <input type="radio"/> EXTERNAL	
	Switch to capacity provider strategy	
Task Definition	<div>Family <input type="text" value="ec2EcsTask"/></div> <div>Revision <input type="text" value="1 (latest)"/></div> <div><input type="button" value="Enter a value"/></div>	
Cluster	<input type="text" value="ecsEc2"/>	
Number of tasks	<input type="text" value="1"/>	
Task Group	<input type="text"/>	

Y si accedemos a la instancia que tenemos asociada a nuestro Cluster, veremos como funciona perfectamente y se muestra nuestra web.



ESTA ES MI PAGINA WEB DE DOCKER

Actividad 4:

Desplegar el contenedor usando Fargate y comparar la experiencia

Finalmente, queda desplegar el contenedor usando Fargate y comparar la experiencia

1. Primero debemos crear un *'Task Definition'*. Poniéndole su nombre, seleccionando su memoria, añadiéndole los roles de **LabRole** y añadiendo el contenedor, usando la URI de nuestro repositorio ECS.

Task definition configuration

Task definition family [Info](#)

Specify a unique task definition family name.

cn_ecs_task

Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

▼ Infrastructure requirements

Specify the infrastructure requirements for the task definition.

Launch type [Info](#)

Selection of the launch type will change task definition parameters.

☒ AWS Fargate

Serverless compute for containers.

☐ Amazon EC2 Instances

Self-managed infrastructure using Amazon EC2 instances.

OS, Architecture, Network mode

Network mode is used for tasks and is dependent on the compute type selected.

Operating system/Architecture [Info](#)

Linux/X86_64

Network mode [Info](#)

awsvpc

Task size [Info](#)

Specify the amount of CPU and memory to reserve for your task.

CPU

1 vCPU

Memory

2 GB

▼ Task roles - conditional

Task role [Info](#)

A task IAM role allows containers in the task to make API requests to AWS services. You can create a task IAM role from the [IAM console](#).

LabRole

Task execution role [Info](#)

A task execution IAM role is used by the container agent to make AWS API requests on your behalf. If you don't already have a task execution IAM role created, we can create one for you.

LabRole

Container – 1 [Info](#)

Essential container

Remove

Container details

Specify a name, container image and whether the container should be marked as essential. Each task definition must have at least one essential container.

Name

webserver_cn

Image URI

329187668450.dkr.ecr.us-east-1.amazonaws.com/webse

Essential container

Yes

Private registry [Info](#)

Store credentials in Secrets Manager, and then use the credentials to reference images in private registries.

☐ Private registry authentication

Port mappings [Info](#)

Add port mappings to allow the container to access ports on the host to send or receive traffic. Any changes to port mappings configuration impacts the associated service connect settings.

Container port

80

Protocol

TCP

Port name

webserver_cn-80-tcp

App protocol

HTTP

Remove

Add port mapping

2. A continuación, creamos nuestro cluster usando **FarGate**. Solo es necesario asignarle un nombre y seleccionar en Infrastucture, el uso de AWS Fargate (Serverless)

Cluster configuration

Cluster name

cluster_ECR

There can be a maximum of 255 characters. The valid characters are letters (uppercase and lowercase), numbers, hyphens, and underscores.

Default namespace - *optional*

Select the namespace to specify a group of services that make up your application. You can overwrite this value at the service level.

cluster_ECR

▼ **Infrastructure** Info Serverless

Your cluster is automatically configured for AWS Fargate (serverless) with two capacity providers. Add Amazon EC2 instances, or external instances using ECS Anywhere.

☒ **AWS Fargate (serverless)**
Pay as you go. Use if you have tiny, batch or burst workloads or for zero maintenance overhead. The cluster has Fargate and Fargate Spot capacity providers by default.

☐ **Amazon EC2 Instances**
Manual configurations. Use for large workloads with consistent resource demands.

☐ **External instances using ECS Anywhere**
Manual configurations. Use to add data centre compute.

3. Seguimos añadiendo el 'Task' a nuestro cluster de ECR.
En configuración de despliegue, seleccionamos 'Task' y en Familia nuestro **Task Definition**.

Deployment configuration

Application type [Info](#)
Specify what type of application you want to run.

☐ **Service**
Launch a group of tasks handling a long-running computing work that can be stopped and restarted. For example, a web application.

☒ **Task**
Launch a standalone task that runs and terminates. For example, a batch job.

Task definition
Select an existing task definition. To create a new task definition, go to [Task definitions](#).

☐ **Specify the revision manually**
Manually input the revision instead of choosing from the 100 most recent revisions for the selected task definition family.

Family **Revision**

cn_ecs_task 1 (LATEST)

Desired tasks
Specify the number of tasks to launch.

1

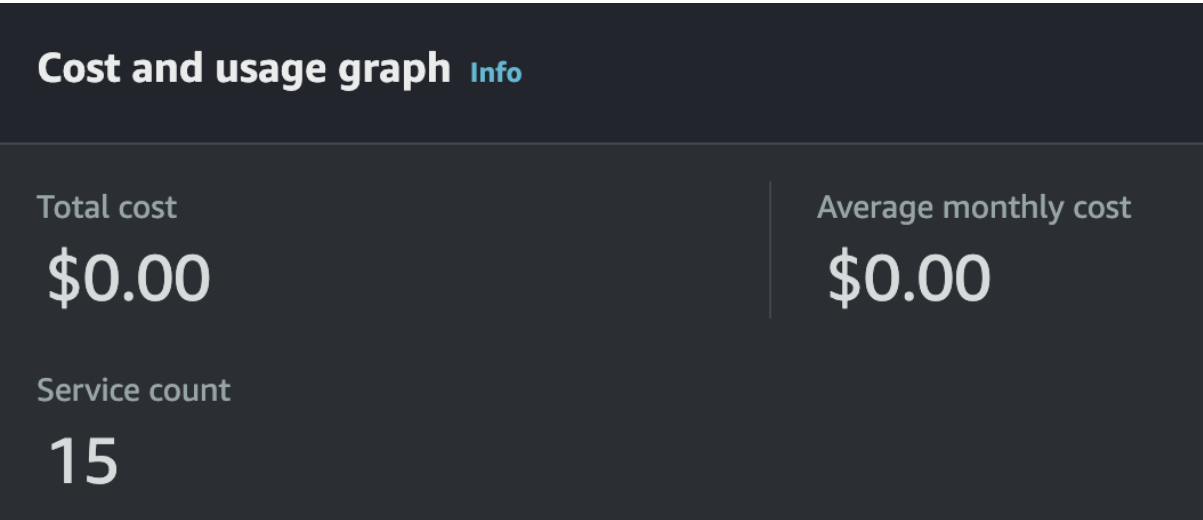
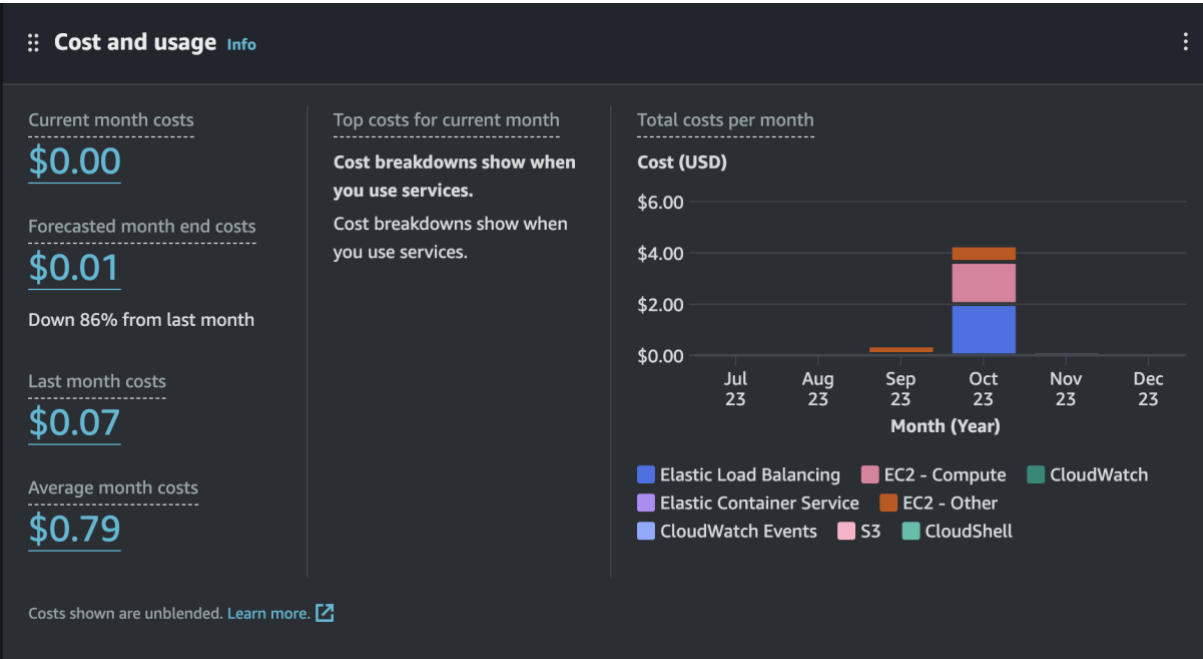
Task group
All tasks with the same task group name are considered as a set when performing spread placement.

Seleccionamos un '**Security Group**' que permita el acceso web -> HTTP(80) desde 0.0.0.0/0

4. Finalmente comprobamos que funciona accediendo a el apartado '**Networking**' de nuestro '**Task**' y seleccionando su IP Pública:



Presupuesto y estimación de gasto de los recursos desplegados



EC2 Container Registry (ECR)	\$0.00
Elastic Container Service	\$0.00
EC2-Instances	\$0.00
EC2 Container Registry (ECR)	\$0.00

